上級編
第 39 講 サイコロの出目

例題

六面のサイコロをある回数振った場合，それぞれの目が出る確率がどのようになるかを検証するシミュレーションプログラムを作成する。
例えば，サイコロを 100,000 回振る場合のシミュレーションを行うと以下のように表示される。

実行結果
```
出目 1 : 16727 回  確率 : 16.727 %
出目 2 : 16497 回  確率 : 16.497 %
出目 3 : 16912 回  確率 : 16.912 %
出目 4 : 16591 回  確率 : 16.591 %
出目 5 : 16588 回  確率 : 16.588 %
出目 6 : 16685 回  確率 : 16.685 %
```

以下のプログラムの空欄に入る最も適切なものを，あとの①～④のうちから一つ選びなさい。

```
import random

kaisuu = 100000 # サイコロを振る回数
Deme_kaisuu = [0, 0, 0, 0, 0, 0] # 出目の回数を格納する
kakuritsu = 0 # 各出目が出る確率を格納する

for i in range(kaisuu) :
  deme = random.randint(1, 6)
  [                    ]

# 結果を表示
for i in range(6):
  kakuritsu = Deme_kaisuu[i] / kaisuu * 100
  print(' 出目 ', i + 1, ':', Deme_kaisuu[i], ' 回 ',  ' 確率 :', kakuritsu, '% ')
```

① `Deme_kaisuu[deme] = Deme_kaisuu[deme] + 1`

② `Deme_kaisuu[deme - 1] = Deme_kaisuu[deme - 1]`

③ `Deme_kaisuu[deme - 1] = Deme_kaisuu[deme - 1] + 1`

④ `Deme_kaisuu[deme - 1] = Deme_kaisuu[deme - 1] - 1`

上級編 第39講 確認問題

六面のサイコロを２個同時に振ったとき，出目の合計が８になる確率を検証するシミュレーションプログラムを作成する。
例えば，100,000 回試行するシミュレーションを行うと，以下のように表示される。

実行結果

```
出目の合計 8 : 13831 回  確率 : 13.831 %
```

以下のプログラムの空欄に入る最も適切なものを，あとの①～④のうちから一つ選びなさい。

```
import random

kaisuu = 100000 # サイコロを振る回数
deme_kaisuu = 0 # 出目の回数を格納する
goukei = 8 # 確率を求めたい合計値（今回は 8）
kakuritsu = 0 # 出目の合計値が出る確率を格納する

for i in range(kaisuu):
  deme1 = random.randint(1, 6)
  deme2 = random.randint(1, 6)
  if (deme1 + deme2) == goukei:
    deme_kaisuu = deme_kaisuu + 1

[                    ]

# 結果を表示
print('出目の合計', goukei, ':', deme_kaisuu, '回',
'確率:', kakuritsu, '%')
```

① kakuritsu = deme_kaisuu / kaisuu

② kakuritsu = kaisuu / deme_kaisuu * 100

③ deme_kaisuu = kakuritsu / kaisuu * 100

④ kakuritsu = deme_kaisuu / kaisuu * 100

上級編　例題　解答

第34講　コインの枚数

例題　解答：④

第35講　駐車料金の判定

例題　解答：④

第36講　割引のある料金

例題　解答：④

第37講　テストの集計

例題　解答：①

第38講　じゃんけん

例題　解答：①

第39講　サイコロの出目

例題　解答：③

第40講　中央値の判定

例題　解答：①

第41講　素数判定

例題　解答：③

第42講　素因数分解

例題　解答：②

第43講　倍数の判定・FizzBuzz

例題　解答：④

第44講　バブルソート・選択ソート

例題　解答：③

第45講　グラフ理論

例題　解答：②

第46講　待ち行列

例題　解答：②

第47講　パリティチェック

例題　解答：①

第48講　すごろくゲーム

例題　解答：④

上級編　第 39 講　確認問題　解答

正解：④

解説：

8 行目の for 文では，設定した試行回数分，出目 1 と出目 2 の値をランダムで生成する繰り返し処理を行っている。

11 行目からの if 文では，シミュレーション中に，出目 1 と出目 2 の合計が 8 となったとき，変数 deme_kaisuu に 1 を加算している。そのため, 100,000 回のシミュレーションが終わったとき（8 行目からの for ブロックを抜けたとき）, 変数 deme_kaisuu には, 出目の合計が 8 になった回数が代入されているといえる。

したがって，14 行目では,「出目の合計が 8 になった回数 ÷ 試行回数 × 100」となる式を当てはめればよいことがわかる。よって，正解は④ `kakuritsu = deme_kaisuu / kaisuu * 100` である。

```
01 import random
02
03 kaisuu = 100000 # サイコロを振る回数
04 deme_kaisuu = 0 # 出目の回数を格納する
05 goukei = 8 # 確率を求めたい合計値（今回は 8）
06 kakuritsu = 0 # 出目の合計値が出る確率を格納する
07
08 for i in range(kaisuu):
09   deme1 = random.randint(1, 6)
10   deme2 = random.randint(1, 6)
11   if (deme1 + deme2) == goukei:
12     deme_kaisuu = deme_kaisuu + 1
13
14 kakuritsu = deme_kaisuu / kaisuu * 100
15
16 # 結果を表示
17 print('出目の合計', goukei, ':', deme_kaisuu, '回',
   '確率:', kakuritsu, '%')
```

実行結果

```
出目の合計 8 : 13936 回　確率: 13.936000000000002 %
```